



TITLE:

Risa / Asir の行列演算と改良について (数式処理 : その研究と目指すもの)

AUTHOR(S):

兵頭, 礼子; 近藤, 祐史; 村尾, 裕一; 齋藤, 友克

CITATION:

兵頭, 礼子 ...[et al]. Risa / Asir の行列演算と改良について (数式処理 : その研究と目指すもの). 数理解析研究所講究録 2012, 1785: 162-166

ISSUE DATE:

2012-03

URL:

<http://hdl.handle.net/2433/172729>

RIGHT:

Risa/Asir の行列演算と改良について

兵頭礼子 近藤祐史 村尾裕一
(株) アルファオメガ 香川高等専門学校 電気通信大学
齋藤友克
(株) アルファオメガ

1 はじめに

数式処理システム Risa/Asir には、行列の乗算アルゴリズムとして、内積アルゴリズムと Strassen-Winograd アルゴリズムが実装されている。Strassen-Winograd アルゴリズムでは、行列の成分として多項式を持ち、また多項式が密なものに対しては内積アルゴリズムよりも計算量の比率以上に高速に演算を行うことができるが、演算速度の大きな要因となるものは、演算の中で使用するメモリ使用量に依存するため、疎な多項式を要素に持つ行列では、Strassen-Winograd アルゴリズムでは内積アルゴリズムと同程度の演算速度となる [14]。そのため、Risa/Asir 上では、標準では、内積アルゴリズムが稼動するようになっており、`ctrl("StrassenSize", n)` としたときのみ、行列サイズが $n \times n$ 以下になるまで Strassen-Winograd アルゴリズムを適用し、計算するように実装されている。また、アルゴリズムの実装だけでなく、疎行列を扱うのに適したデータ構造を検討、実装実験を行い、演算時間の比較実験から、数式処理に適したデータ型の考察を行なっている ([10],[11],[12],[13])。今回、Risa/Asir で現行の CPU を使った行列の乗算の実験を行い、その結果報告と、逆行列の演算の実装について報告する。

2 行列の乗算実験

実験対象行列として、 $n \times n$ であり各成分は表 1 のような成分を持つ行列 A, B を考える。これらの行列を

表 1: 実験データ

Case I	$a_{ij} = (i+1)(j+1)(x^5 + x^4 + x^3 + x^2 + x)$ $b_{ij} = (i+1)(j+1)(y^5 + y^4 + y^3 + y^2 + y)$
Case II	$a_{ij} = x^{(2*i+1)} + x^i + x^{(j+1)} + i \cdot j$ $b_{ij} = x^{(2*i+1)} + x^i + x^{(j+1)} + i \cdot j$

使い、Risa/Asir 上で内積アルゴリズムと Strassen-Winograd アルゴリズムを適用して $A \cdot B$ を実行し、演算時間を計測する。使用するマシンは CPU : AMD Phenom 9750 X4(2400MHz)、メモリ : 4GB、FreeBSD 8.2 Stable amd64 上で実験を行った。

2.1 演算時間と考察

表 1 の行列を用いて、Risa/Asir 上で内積アルゴリズムと Strassen-Winograd アルゴリズムでのそれぞれの演算時間を表 2、表 3 に示す。また、同様の実験を CPU : AMD AthlonXP 3200+(2200MHz)、メモリ : 512MB、FreeBSD 5.2 Stable 上で実験を行った結果を表 4、表 5 に示す [14]。

表 2: Case I の演算時間の比較 (単位: 秒)

Size n	内積アルゴリズム			Strassen-Winograd algorithm		
	CPUtime	GC	total	CPUtime	GC	total
8	0.008658	0.007521	0.02007	0.005361	0.003412	0.008773
16	0.07726	0.09605	0.1645	0.02084	0.0288	0.04821
32	0.6552	0.8017	1.458	0.1255	0.1207	0.2457
64	5.468	7.25	12.72	0.5941	0.5475	1.141
128	45.48	73.54	119	2.907	3.153	6.06
256	375.8	682.2	1060	13.76	16.72	30.48

表 3: Case II の演算時間の比較 (単位: 秒)

Size n	内積アルゴリズム			Strassen-Winograd algorithm		
	CPUtime	GC	total	CPUtime	GC	total
8	0.001905	0.007272	0.003724	0.001042	0.000716	0.003517
16	0.02153	0.01992	0.04144	0.02082	0.0576	0.02658
32	0.2368	0.2642	0.5008	0.2079	0.04662	0.2545
64	2.492	3.91	6.402	2.004	0.5793	2.584

表 4: AthlonXP 上での Case I の演算時間の比較 (単位: 秒)

Size n	内積アルゴリズム			Strassen-Winograd algorithm		
	CPUtime	GC	total	CPUtime	GC	total
8	0.008180	0.004957	0.01351	0.008301	0.005341	0.01408
16	0.05496	0.03483	0.09006	0.04470	0.02531	0.07020
32	0.4802	0.3173	0.8023	0.2416	0.1718	0.4143
64	4.068	3.177	7.288	1.215	1.014	2.237
128	35.76	29.08	65.12	6.007	4.866	10.94
256	307.4	311.2	622.2	28.88	28.76	58.06

Strassen-Winograd アルゴリズムの適用で、AthlonXP では十分な速度を得られなかった Case II の場合でも、現行の CPU である Phenom を使った場合 (表 3) では十分に高速化されている。CPU 上のキャッシュメモリの増加等、考えられる要因はさまざまであるが、疎な多項式を要素にもつ行列の乗算でも、Strassen-Winograd アルゴリズムの適用によって十分高速化されていると考えられる。アーキテクチャや要素となる多項式・行列の要素の密度によって同じアルゴリズム、同じ要素でも演算速度が変わっており、今度もアー

表 5: AthlonXP 上での Case II の演算時間の比較 (単位: 秒)

Size n	内積アルゴリズム			Strassen-Winograd algorithm		
	CPUtime	GC	total	CPUtime	GC	total
8	0.005243	0.002421	0.008109	0.005653	0.002519	0.008416
16	0.05067	0.02646	0.07772	0.04965	0.02825	0.07849
32	0.5060	0.2568	0.7791	0.4887	0.2456	0.7359
64	5.382	3.623	9.046	4.821	3.174	8.040

キテクチャ別、要素別の実験を続け、アルゴリズム選択のベンチマークとなるような要素の検討などを行いたい。

3 逆行列

Risa/Asir には、逆行列の演算では、ガウスの消去法アルゴリズムが実装されている。今回、Strassen の逆行列アルゴリズムを実装し、演算時間の比較を行った。

3.1 Strassen のアルゴリズム

行列 A を

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} = \begin{pmatrix} I & 0 \\ A_{21}A_{11}^{-1} & I \end{pmatrix} \begin{pmatrix} A_{11} & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} I & A_{11}^{-1}A_{12} \\ 0 & I \end{pmatrix}$$

(ただし $A_{11}^{-1} = A_{11}^{-1}$, $A_{21}A_{11}^{-1}A_{12}$) のようにブロック LDU 分解可能ならば、逆行列は以下のように計算できる。

$$\begin{aligned} A^{-1} &= \begin{pmatrix} I & A_{11}^{-1}A_{12} \\ 0 & I \end{pmatrix} \begin{pmatrix} A_{11} & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} I & 0 \\ A_{21}A_{11}^{-1} & I \end{pmatrix} \\ &= \begin{pmatrix} A_{11}^{-1} + A_{11}^{-1}A_{12}A_{21}A_{11}^{-1} & A_{11}^{-1}A_{12} \\ A_{21}A_{11}^{-1} & A_{11}^{-1} \end{pmatrix} \end{aligned}$$

となる。また、Strassen の方法で逆行列の計算量は、

$$O(7^k) \approx O(7^{\log_2 n}) = O(n^{\log_2 7}) \approx O(n^{2.8})$$

となることが示されている [6]。

このアルゴリズムを、Risa/Asir 上に実装し、逆行列の演算時間の計測を行った。

3.2 実験

実験対象行列として、 $n \times n$ であり各成分は $a_{ij} = \text{random } x$ となる行列 A を使い、演算速度の計測を行った。使用マシンは CPU: AMD Phenom 9750 X4(2400MHz)、メモリ: 4GB、FreeBSD 8.2 Stable amd64 である。表 6 に演算時間を示す。今回の Strassen の逆行列アルゴリズムの適用実験では、ガウスの消去法アルゴリズムに比べ高速演算を行うことができなかった。今後もアルゴリズムの実装方法や要素となる多項式の種類など、検討を重ねていく必要がある。また、行列の乗算の実験同様、アーキテクチャ別、要素別の実験を続け、アルゴリズム選択のベンチマークとなるような要素の検討などを行いたい。

表 6: 演算時間の比較 (単位: 秒)

Size n	ガウスの消去法			Strassen algorithm		
	CPUtime	GC	total	CPUtime	GC	total
8	0.000553		0.0005519	0.000595		0.000596
16	0.006281	0.000946	0.007228	0.005104	0.002763	0.007867
32	0.04237	0.0207	0.06451	0.1494	0.03045	0.1798
64	0.5026	0.2055	0.7085	85.57	11.38	96.96

参 考 文 献

- [1] D. Bini and V.Y. Pan. *Polynomial and Matrix*, Vol. 1 of *Progress in Theoretical Computer Science*. Birkhäuser, 1994.
- [2] D. Coppersmith and S. Winograd. Matrix multiplication via arithmetic progressions. *Journal of Symbolic Computation*, 9:251–280, 1990.
- [3] M. L. Griss. The algebraic solution of sparse linear systems via minor expansion. *ACM Transactions on Mathematical Software*, Vol. 2, No. 1, pp. 31–49, 1976.
- [4] E. Horowitz and S. Sahni. On computing the exact determinant of matrices with polynomial entries. *Journal of the ACM*, Vol. 22, No. 1, pp. 38–50, 1975.
- [5] M. Noro and T. Takeshima. Risa/Asir — a computer algebra system. In P. S. Wang, editor, *Proceedings of ISSAC '92*, pages 387–396, Berkeley, CA, July 27–29 1992.
- [6] V. Strassen. Gaussian elimination is not optimal. *Numer. Math.*, 13:354–356, 1969.
- [7] S. Winograd. On multiplication of 2×2 matrices. *Linear Algebra and its Applications*, 4:381–388, 1971.
- [8] 兵頭 礼子, 村尾 裕一, 齋藤 友克. Risa/asir の matrix 演算の検討. 数式処理, 9:10–11, 2002. 第 11 回大会報告.
- [9] 兵頭 礼子, 村尾 裕一, 齋藤 友克. Risa/asir の matrix 演算の新しい実装について. 講究録 1295 「Computer Algebra – Algorithms, Implementations and Applications, 2001」, pages 213–219. 京都大学数理解析研究所, 2002.
- [10] 兵頭 礼子, 村尾 裕一, 齋藤 友克. 数式処理のための行列演算の効率的な実装法について. 数式処理, 10:18–19, 2003. 第 12 回大会報告.
- [11] 兵頭 礼子, 村尾 裕一, 齋藤 友克. 多項式表現と行列演算の改良. 講究録 1335 「Computer Algebra – Algorithms, Implementations and Applications, 2002」, pages 28–32. 京都大学数理解析研究所, 2003.
- [12] 兵頭 礼子, 村尾 裕一, 齋藤 友克. 行列計算と基本線形演算の実装法について. 講究録 1395 「Computer Algebra – Algorithms, Implementations and Applications, 2003」, pages 218–223. 京都大学数理解析研究所, 2004.
- [13] 兵頭礼子, 村尾裕一, 齋藤友克. Risa/Asir の行列演算の実装 II. 講究録 1456 「Computer Algebra – Design of Algorithms, Implementations and Applications, 2005」, pages 144–149. 京都大学数理解析研究所, 2005.

- [14] 兵頭 礼子, 村尾 裕一, 齋藤 友克. Matrix Multiplication Made Fast — Practical View of Fast Matrix Operation for Computer Algebra System. 数式処理, 11:3–19, 2005. Noda2005 論文.